



Automatic Shape Control of Deformable Wires based on Model-Free Visual Servoing

Romain Lagneau, Alexandre Krupa, Maud Marchal

► To cite this version:

Romain Lagneau, Alexandre Krupa, Maud Marchal. Automatic Shape Control of Deformable Wires based on Model-Free Visual Servoing. IEEE Robotics and Automation Letters, 2020, 5 (4), pp.5252 - 5259. 10.1109/LRA.2020.3007114 . hal-02890830

HAL Id: hal-02890830

<https://inria.hal.science/hal-02890830>

Submitted on 6 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Automatic Shape Control of Deformable Wires based on Model-Free Visual Servoing

Romain Lagneau¹, Alexandre Krupa² and Maud Marchal¹

Abstract—In this paper, we propose a novel approach to automatically control the 3D shape of deformable wires using robots. Our approach proposes a novel visual feature along with a novel shape servoing method to enable dual arm manipulation of deformable wires. The visual feature relies on a geometric B-spline model and the use of Sequential Importance Resampling (SIR) particle filtering to track the 3D deformed shape of a wire over time. The shape servoing method is an adaptive model-free method that iteratively updates the deformation Jacobian matrix using weighted least-squares minimization with sliding window and an eigenvalue-based confidence criterion. We performed several experiments on wires with different mechanical properties. The results show that our approach succeeded to control the 3D shape of various wires for many different desired deformations, while working at an interactive time. It has also been shown that the shape servoing method can be used to handle large deformations by subdividing the task in successive intermediary targets to reach. These promising results pave the way for automatic control of the 3D shapes of deformable wires in many fields such as catheter insertion in medicine or wire manipulation in industry.

I. INTRODUCTION

Deformable wires are used in various fields and the control of their shapes by robots is of interest for many applications, such as in medicine for robotic needle insertion [1] or the control of catheter [2]–[4] or in industry for automatic routing of wires [5]. However, as of today, industrial robots still cannot perform easily such tasks due to the complex behavior of wires subject to deformations.

In this paper, we propose a novel approach to automatically control the 3D shape of deformable wires. The automatic control of the shape of an object will be thereafter denoted *shape servoing*. Our contributions are as follows:

- a novel visual feature that permits to determine at an interactive time the 3D shape of wires using a geometric model that is independent from the mechanical properties of the wire ;
- an adaptive model-free shape servoing method that controls two robot end-effectors to make a wire reach a desired 3D shape moving solely its extremities.

Our approach does not need any mechanical information on the wire nor marker on it to perform the control of the 3D shape. In addition, computation time performances allow its

use at an interactive time for handling various manipulations and transformations, even for large deformations.

The remainder of this paper is organized as follows: Sect. II presents the related work on wire tracking and wire shape control, Sect. III describes the proposed methods while Sect. IV details the experimental evaluation. Finally Sect. V ends the paper with a discussion on the approach.

II. RELATED WORK

Controlling the shape of a deformable wire requires to track its 3D shape at an interactive time and to provide a control law to make it converge towards a desired 3D deformed shape. Therefore, in this section, we first review wire shape control and then wire tracking.

A. Wire shape control

We will divide wire shape control methods into two categories, whether they are using or not mechanical models.

To control the shape of a deformable wire, methods relying on physics-based models have been proposed to predict the behavior of the wire. For instance, Laranjeira *et al.* proposed a visual servoing method to control the slackness of a tether that connects two robots [6]. This method uses a mechanical model based on the catenary analytic formulation to represent the tether. A dead zone compensation and a Coulomb friction compensation models have been employed in a combined force-position control method in [3]. This method permits to make a catheter-like robot capable of coming into contact with a moving target at a desired force. Alternatively, mass-spring models have been used to control the deformation of a wire. For instance, controlling the oscillations that occur when a wire is manipulated at one of its extremities by a gripper has been addressed in [7]. The wire is modeled as a 1D mass-spring model for the oscillations. The oscillation attenuation challenge has also been addressed using the Finite Element Method (FEM) [8]. A local dynamic model of the wire is proposed. This model is used to derive a position-based control law whose stability is ensured using the Lyapunov theory. The method has been tested only by simulations.

To avoid the complexity of accurately modeling the wire deformation behavior, model-free methods have been proposed. For instance, a method to automatically control the shape of a deformable object based on the online estimation of pseudo stiffness matrices has been proposed in [9]. However, the method does not handle well wire manipulation when the wire presents a small bending, nor cases where the feature points are not well spread on the object. A method based on a

This work was supported by the European Unions Horizon 2020 research and innovation program under grant agreement No 731761 for project “Imagine”.

¹Univ. Rennes, INSA, IRISA, Inria

²Univ. Rennes, Inria, IRISA

e-mail: {romain.lagneau, alexandre.krupa, maud.marchal}@irisa.fr.

novel online Gaussian Process Regression has been proposed in [10]. The method requires a preliminary learning phase during which a set of feature points velocities is acquired along with the corresponding manipulated points velocities. This training set can be heavy depending on the manipulation. A learning-based approach relying on the Coherent Point Drift (CPD) method has been used both for wire tracking, task planning and trajectory planning in [11]. The CPD method permits to compute the non-rigid transformation of a point set into another one and to determine if the current tracked state of the wire is identified in a knowledge database of wire predetermined states. If it belongs to the database, the next manipulation command is provided from the database. Other planning methods have been proposed, such as the method of Zhu *et al.* to perform wire-routing tasks [5]. The planner takes both visual and force measurements as inputs. The planner controls in position two effectors: one mobile that moves one extremity of the wire and one that can only release or hold more firmly the second extremity. This approach relies on a pre-selected set of contact points with the environment to achieve a desired configuration. Wire shape control has also been addressed using visual servoing methods. For instance, an adaptive model-free control method for catheter-like soft robot has been proposed in [12]. This method permits to align the eye-in-hand camera with an optical target when the catheter is inserted into a constrained environment. This method relies on several assumptions: viscoelastic and frictional losses are neglected and shearing is ignored. Another method to control a 3-Degree-Of-Freedom (DOF) catheter-like soft robot has been proposed in [13]. This method permits to make the tip of the catheter converge towards a desired position. To do this, six arbitrary operating points are selected anywhere on the catheter. For each of them, a local controller computes the control law to achieve this objective. A fuzzy controller mixes the results of the six controllers to generate the global control law. The system has been tested on Matlab simulation only. More recently, a method to control the 2D shape of a wire using several effectors has been proposed by Zhu *et al.* [14]. Each effector has 3 DOFs (2 translations, 1 rotation). The wire shape is approximated by a Fourier series. The geometric deformation model is adapted online, assuming that a change in the Fourier coefficients depends linearly on a change in robot position. Sets of changes in Fourier coefficients and sets of changes in robot position are continuously acquired. These sets are used in a least-square minimization to compute the Jacobian matrix that relates the robot motions to the changes of shape. However, if the robot did not move in all the affordable directions, the Jacobian matrix estimation update is delayed to avoid singularity. The Jacobian matrix is finally used to compute the control law. The method was tested using a wire held by two grippers.

B. Wire tracking

In order to control the deformation of a wire, it is necessary to be able to track its shape. Matsuno *et al.* proposed a method that focuses on a particular case of deformation: the knotted wire [15]. The method uses a topological model that relies on

knot theory in order to represent a wire. The use of invariants of knots allows to assess whether two knots are similar or not.

Image processing techniques have been used to track a deformable wire in the image plane in [16]. In the initialization phase, the user must select a seed point in a region of the image that corresponds to one extremity of the wire. Some characteristics of this region are extracted, such as luminance or global orientation. Then, the method scans the whole image region by region to detect the global shape of the wire by finding a region with similar characteristics.

More recently, Padoy *et al.* proposed a wire tracking method based on texture augmentation of the wire [17]. The wire texture is augmented by adding a pattern composed of alternating colors. An energy minimization method is used to adapt a NURBS spline model, representing the 3D configuration of the wire, with the visual information captured by calibrated stereo cameras.

Probabilistic methods have been proposed for deformable object tracking. For instance, a probabilistic framework working with physical simulations to track deformable wires using point clouds is proposed in [18]. Consequently, knowledge of the physical properties of the object is required. Another method is proposed in [19], which consists of a probabilistic registration method that uses visibility information for not considering occluded model points for the registration. An history of previously faced situations is saved in order to perform a k-Nearest Neighbors (k-NN) template recovery when tracking failures occur. Finally, a probabilistic approach that considers a physical model of the object has been proposed in [20] for tracking deformable objects from point clouds.

C. Our contributions

Our contributions to the automatic shape control of deformable wires are as follows: the definition of a new visual feature permitting to model the 3D shape of a wire and a novel shape servoing method to control its deformations. Unlike the tracking methods mentioned above, such as [18] or [20], the visual feature that we propose does not require any marker nor any mechanical model to infer the 3D shape of a wire. Instead, it relies on a geometric model based on B-splines [21] combined with the use of Sequential Importance Resampling (SIR) particle filtering [22]. The use of B-splines allows our method controlling the 3D shape of a wire, while the method proposed in [14] handled only 2D shapes. Moreover, it permits to handle small bending situations and to ensure having equally spread feature points on the length of the object, contrarily to the method proposed in [9]. The use of SIR particle filtering permits to determine which B-spline best matches the visual observations provided by a RGB-D camera. The shape control is carried out by a closed-loop visual servoing. Two end-effectors holding the wire extremities are controlled in order to make the wire converge towards a desired 3D shape. The control law is computed using a model-free shape servoing method inspired from [23]. Contrarily to [10], the shape servoing method that we propose does not require a complex initialization phase. The Jacobian matrix that relates the motions of the end-effectors to the deformation

of the wire is initialized by small motions in every direction the system can afford and is adapted online using weighted least-squares minimization with sliding window. A confidence threshold permits to update solely the parts of the Jacobian matrix impacted by the most recent directions of motion of the robots, while the method proposed in [14] keeps the Jacobian matrix constant when the robots have not been moved in all the possible directions.

III. METHODOLOGY

The challenge addressed in this paper is the automatic 3D shape control of deformable wires. To address this challenge, we first propose a novel visual feature that permits to infer the 3D shape of a deformable wire based on particle filtering. Secondly, we propose a visual servoing approach to control two end-effectors holding the extremities of the wire. This approach has the advantage of not relying on any a priori mechanical model.

In this section, we will first explain the novel visual feature. Then, we will present the active shape servoing method.

A. Visual feature for deformable wire

We developed a novel visual feature that permits to infer the 3D shape of a deformable wire from a point cloud using a RGB-D camera. The shape of a wire is represented by a geometric model based on B-splines. A B-spline is a piecewise polynomial function with a user-defined number of segments and order θ . A B-spline can be constructed from a set of control points \mathbf{c} , whose number depends on the number of segments and the order of the B-spline.

We propose to use a SIR particle filter to track the wire. A particle filter generates χ particles, also known as sample states, representing the internal state of a system that is observed. At each time step, it updates each sample state according to a predictive update rule. Once updated, the likelihood between each sample state and the observations of the system is computed. Finally, a weighted sum of all the sample states is performed to generate an estimate of the internal state of the observed system. The weight of a sample is proportional to its likelihood.

In our case, the SIR filter sample states are B-splines and the observation is a segmented 3D point cloud of the wire. Let ψ be a B-spline of the SIR filter, $\tilde{\mathbf{p}} \in \mathbb{R}^3$ a point of a B-spline, η the number of equidistant points used to model the wire and \mathbf{p} an observed point of the wire in the point cloud.

1) *Update of a particle:* Let \mathbf{m}_{ac_j} and \mathbf{m}_{bc_j} be the predictive motions of an arbitrary control point \mathbf{c}_j of a B-spline that result respectively from the motion of each end-effector. A prediction of these motions can be defined by:

$$\begin{aligned}\mathbf{m}_{ac_j} &= (\mathbf{v}_a + (\mathbf{c}_j - \tilde{\mathbf{p}}_1) \wedge \boldsymbol{\omega}_a) \Delta t \\ \mathbf{m}_{bc_j} &= (\mathbf{v}_b + (\mathbf{c}_j - \tilde{\mathbf{p}}_\eta) \wedge \boldsymbol{\omega}_b) \Delta t\end{aligned}$$

where:

- $\mathbf{v} = (v_x, v_y, v_z)^T$ denotes the linear velocity of an end-effector while $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$ denotes its angular

velocity. The subscript "a" represents the first end-effector and the "b" the other one,

- $\mathbf{a} \wedge \mathbf{b}$ is the vectorial product between two vectors,
- Δt is the robot control law sample time.

The update rule that we propose to compute the new position of each control point \mathbf{c}_j of a B-spline model is as follows:

$$\begin{aligned}\mathbf{c}_j(k) &= \mathbf{c}_j(k-1) + (\mathbf{t}_{c_j} \cdot \mathbf{m}_{ac_j}) \frac{\theta-j}{\theta} \mathbf{m}_{ac_j} \\ &\quad + (\mathbf{t}_{c_j} \cdot \mathbf{m}_{bc_j}) \frac{j}{\theta} \mathbf{m}_{bc_j} + \mathbf{v}\end{aligned}\quad (1)$$

where:

- k is the current discrete time,
- \mathbf{t}_{c_j} is the tangent of the B-spline at the control point \mathbf{c}_j ,
- \mathbf{v} is a vector of Gaussian noise with a zero mean and a standard deviation σ_v ,
- θ is the order of the B-spline,
- $\mathbf{a} \cdot \mathbf{b}$ is the scalar product between two vectors.

The whole B-spline model is updated accordingly to the new position of its control points.

2) *Likelihood evaluation:* Let $\mathbf{p}_{closest}$ be an observed point in the point cloud that is the closest from a B-spline point $\tilde{\mathbf{p}}$. The likelihood of a B-spline point $w(\tilde{\mathbf{p}})$ is evaluated on a window of the depth map taking into account $\Omega \times \Omega$ neighboring points, as depicted in Fig. 1. The further a neighboring point is from the evaluated point $\tilde{\mathbf{p}}$ the smaller is its weight in the likelihood computation.

Let $\mathbf{G} \in \mathbb{R}^{\Omega \times \Omega}$ be the Gaussian weight matrix of standard deviation σ_w defined by:

$$G_{i,j} = \frac{1}{2\pi\sigma_w^2} \exp - \frac{(i-1-A)^2 + (j-1-A)^2}{A^2} \quad (2)$$

where $A \in \mathbb{N}$ such as $A \leq \frac{\Omega}{2} < A+1$.

Let g be a Gaussian function of standard deviation σ_g that represents the similarity between a B-spline point $\tilde{\mathbf{p}}_i$ and its closest observed point $\mathbf{p}_{closest}$. It is defined by:

$$g(\tilde{\mathbf{p}}, \mathbf{p}_{closest}) = \frac{1}{\sigma_g \sqrt{2\pi}} \exp - \frac{\|\tilde{\mathbf{p}} - \mathbf{p}_{closest}\|^2}{2\sigma_g^2} \quad (3)$$

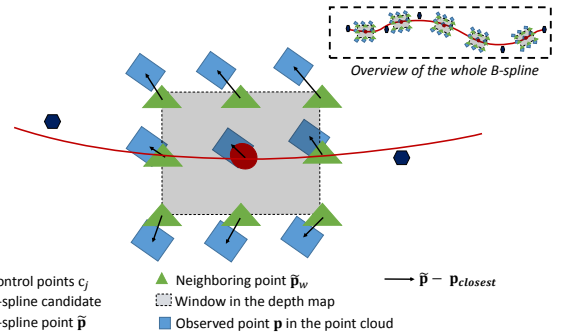


Fig. 1: 2D illustration of the likelihood computation.

The likelihood of a B-spline point $w(\tilde{\mathbf{p}})$ is defined by:

$$w(\tilde{\mathbf{p}}) = \frac{1}{\sum_{i,j=1}^{\Omega} G_{i,j}} \sum_{i,j=1}^{\Omega} G_{i,j} g(\tilde{\mathbf{p}}_w(i, j), \mathbf{p}_{closest}) \quad (4)$$

$$\text{with } \tilde{\mathbf{p}}_w(i, j) = \tilde{\mathbf{p}} + 2\sigma_w \left(\frac{i-1}{A} - 1 \right) \mathbf{t}_{\tilde{\mathbf{p}}} + 2\sigma_w \left(\frac{j-1}{A} - 1 \right) \mathbf{n}_{\tilde{\mathbf{p}}}$$

where $\mathbf{t}_{\tilde{\mathbf{p}}}$ is the tangent of the B-spline at the wire point and $\mathbf{n}_{\tilde{\mathbf{p}}}$ is the normal, located in the XZ plane of the depth camera, of the B-spline at the wire point.

The likelihood of a B-spline $W(\psi)$ is defined as:

$$W(\psi) = \frac{\sum_{i=1}^{\eta} w(\tilde{\mathbf{p}}_i)}{\eta} \quad (5)$$

We obtain the optimal B-spline that represents the current state of the wire resulting from the SIR filter $\hat{\psi}$ by:

$$\hat{\psi} = \frac{1}{\sum_{i=1}^{\chi} W(\psi_i)} \sum_{i=1}^{\chi} W(\psi_i) \psi_i = (\hat{\mathbf{p}}_1, \dots, \hat{\mathbf{p}}_{\eta}) \quad (6)$$

B. Adaptive shape servoing method

Let \mathbf{s} be the current deformation feature vector that represents the 3D shape of a wire and \mathbf{s}^* be the desired one. In this paper, we propose to use η equidistant B-spline points to construct the deformation feature vector \mathbf{s} used in the control law. The deformation feature vector is defined as follows:

$$\mathbf{s} = (\hat{\mathbf{p}}_{1x}, \hat{\mathbf{p}}_{1y}, \hat{\mathbf{p}}_{1z}, \dots, \hat{\mathbf{p}}_{\eta x}, \hat{\mathbf{p}}_{\eta y}, \hat{\mathbf{p}}_{\eta z})^T \in \mathbb{R}^{3\eta} \quad (7)$$

Let define m as the sum of the number of DOFs of the end-effectors, which is 12 in our case since two 6-DOF robotic arms are used to manipulate the wire. Let \mathbf{J} denote the deformation Jacobian that relates the evolution of the 3D shape of the wire with the end-effectors velocity $\mathbf{v} = (\mathbf{v}_a^T, \boldsymbol{\omega}_a^T, \mathbf{v}_b^T, \boldsymbol{\omega}_b^T)^T$ such that $\dot{\mathbf{s}} = \mathbf{J}\mathbf{v}$. In our shape servoing method, we propose to online estimate the deformation Jacobian matrix. By considering that small motion will be applied by the end-effectors between successive control law sample time Δt , the variation of the deformation feature vector $\dot{\mathbf{s}}$ is linear with regards to the motions of the end-effectors. This relation can be written as:

$$\dot{s}_i = \hat{\mathbf{J}}_{i,:} \mathbf{v} + v_i \in \mathbb{R} \text{ for } i \in \{1, \dots, 3\eta\} \quad (8)$$

where \dot{s}_i is the i^{th} element of the variation of the deformation feature vector, the vector $\hat{\mathbf{J}}_{i,:}$ corresponds to the i^{th} row of the estimated Jacobian matrix $\hat{\mathbf{J}}$ and v_i is the i^{th} element of the measurement noise vector.

To determine the Jacobian matrix that respects (8) for the current and previous measurements, we propose to use a weighted least-square minimization with sliding window method. Let define the quadratic cost-function $C(\hat{\mathbf{J}}_{i,:})$ by:

$$C(\hat{\mathbf{J}}_{i,:}) = \sum_{j=k-N}^k \left(\frac{\lambda^{k-j}}{m^2(j-1)} (\dot{s}_i(j) - \hat{\mathbf{J}}_{i,:} \mathbf{v}(j-1))^2 \right) \in \mathbb{R} \quad (9)$$

where N is the size of the sliding window, $0 < \lambda \leq 1$ is a constant forgetting factor giving less importance to the oldest measures, $\dot{s}_i(j)$ are the variations of the deformation feature vector at the previous discrete times j and $m(k) = \sqrt{1 + \mathbf{v}^T(k)\mathbf{v}(k)}$ is a normalization signal.

Let define:

$$\mathbf{R} = \sum_{j=k-N}^k \left(\frac{\lambda^{k-j}}{m^2(j-1)} \mathbf{v}(j-1) \mathbf{v}^T(j-1) \right) \in \mathbb{R}^{m,m} \quad (10)$$

$$\mathbf{Q} = \sum_{j=k-N}^k \left(\frac{\lambda^{k-j}}{m^2(j-1)} \dot{s}_i(j) \mathbf{v}(j-1) \right) \in \mathbb{R}^{m,1} \quad (11)$$

The cost function is convex with regards to $\hat{\mathbf{J}}_{i,:}^T$, thus its minimum is obtained for the value of $\hat{\mathbf{J}}_{i,:}^T$ that nullifies its gradient ∇C , where $\nabla C = -2\mathbf{Q} + 2\mathbf{R}\hat{\mathbf{J}}_{i,:}^T$.

The estimated Jacobian matrix row that nullifies the gradient can be written as follows:

$$\hat{\mathbf{J}}_{i,:}^T = \mathbf{R}^{-1} \mathbf{Q} \in \mathbb{R}^{m,1} \quad (12)$$

One can notice that the eigenvalues of the matrix \mathbf{R} tend towards 0 when the system converges to the desired shape. Therefore, directly inverting the matrix would induce instability in the Jacobian matrix estimation. To avoid this problem, we propose to use the following confidence criterion based on eigenvalue decomposition in the Jacobian estimation. The matrix \mathbf{R} being a positive-definite symmetric $m \times m$ matrix, it can be decomposed such as:

$$\begin{aligned} \mathbf{R} &= \boldsymbol{\Phi} \boldsymbol{\Gamma} \boldsymbol{\Phi}^T \\ \text{with } \boldsymbol{\Phi} &= [\boldsymbol{\phi}_1 \quad \dots \quad \boldsymbol{\phi}_m] \in \mathbb{R}^{m,m} \\ \text{and } \boldsymbol{\Gamma} &= \begin{bmatrix} \gamma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \gamma_m \end{bmatrix} \in \mathbb{R}^{m,m} \end{aligned} \quad (13)$$

where $\gamma_1 \geq \dots \geq \gamma_m \geq 0$ are the eigenvalues of \mathbf{R} and $\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_m$ are its eigenvectors. Finally, the new estimate of the deformation Jacobian is computed depending on a user-defined confidence threshold ϵ :

$$\hat{\mathbf{J}}_{i,:}^T(k) = \begin{cases} \mathbf{R}^{-1} \mathbf{Q} & \text{if } \gamma_1 > \epsilon, \dots, \gamma_m > \epsilon \\ \hat{\mathbf{J}}_{i,:}^T(k-1) & \text{if } \gamma_1 \leq \epsilon, \dots, \gamma_m \leq \epsilon \\ \mathbf{V}_1 \mathbf{Q} + \mathbf{V}_2 \hat{\mathbf{J}}_{i,:}^T(k-1) & \text{otherwise} \end{cases} \quad (14)$$

$$\text{with } \mathbf{V}_1 = \boldsymbol{\Gamma}_{:,1:j} \boldsymbol{\Phi}_{1:j,1:j}^{-1} \boldsymbol{\Gamma}_{:,1:j}^T$$

$$\text{and } \mathbf{V}_2 = \boldsymbol{\Gamma}_{:,j+1:m} \boldsymbol{\Gamma}_{:,j+1:m}^T$$

with ϕ sorted such as $\phi_j > \epsilon$ and $\phi_{j+1} \leq \epsilon$

The subscript “:” indicates that all the rows (respectively columns) are selected when it appears before (respectively after) the comma, while the subscript “1:j” indicates that all the rows (or columns) from index 1 to index j are selected. The \mathbf{V}_1 term in (14) allows to update the Jacobian in the directions where the eigenvalue-based confidence criterion is met, while the term \mathbf{V}_2 allows to ignore the information in the noisy directions.

The velocity control law finally applied to control the two end-effectors is defined by $\mathbf{v}^{ctrl} = -\alpha \hat{\mathbf{J}}^+ (\mathbf{s} - \mathbf{s}^*) \in \mathbb{R}^m$ where $\alpha \in \mathbb{R}^+$ denotes the control law gain and $\hat{\mathbf{J}}^+$ denotes the Moore-Penrose pseudo-inverse of the estimated deformation Jacobian $\hat{\mathbf{J}}$.

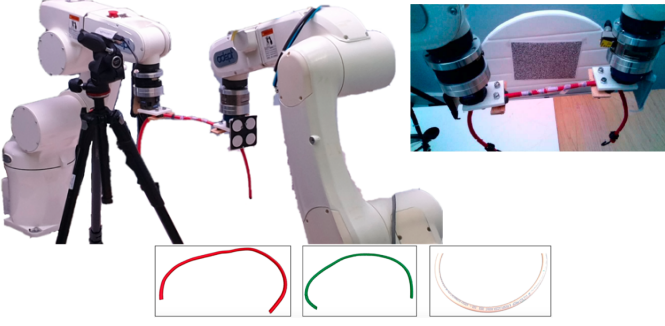


Fig. 2: Experimental setup composed of two robot arms that manipulate a wire tracked by a RGB-D camera. (top-right) Resting state common to every experiments. (bottom) The different types of wires that were tested.

IV. EXPERIMENTS

A. Experimental Setup

The experimental setup, shown in Fig.2, is composed of two 6-DOF anthropomorphic robot arms (Viper 650 and 850 from ADEPT). Their end-effectors permit to attach a deformable wire. The length of the deformable wires section that is considered during the experiments is around 150mm.

A remote Intel Realsense D435 RGB-D camera is used for tracking the deformable wire. For all the experiments, the wire was placed approximately at 400mm of the RGB-D camera, so the depth information is accurate within $\pm 0.6\text{mm}$ [24]. To improve the accuracy of the depth measurements, the texture of the wire has been augmented by randomly sticking some colored pieces of paper. These are not used for the tracking. Color and depth segmentation are performed using PCL library [25] to isolate the wire from the rest of the observed point-cloud.

The homogeneous transformation matrix between the camera coordinates frame and the robots fixed coordinates frames is computed during an initialization step using markers stuck on the end-effectors thanks to ViSP library [26].

To define a desired wire shape, the robots are moved before an experiment and the resulting shape is recorded. The robots and the wire are thereafter put back in their resting state, common to every experiments and depicted in Figure 2.

To initialize the deformation Jacobian, an open-loop initial procedure is performed where both end-effectors move in every possible direction successively on a very short distance while measuring the resulting deformation variation of the wire. The initial values of the components of the deformation Jacobian are then numerically computed by finite difference. Since this initial excitation is performed for each DoF of the system, the initial value of the deformation Jacobian is therefore full rank. Moreover, it will maintain full rank during the shape visual control thanks to the Jacobian update (14) that is performed only along with the sufficiently excited directions.

For all the experiments, the parameters were set empirically as follows: α = between 0.0125 and 0.025, $\lambda=0.99$, $N=10$ and $\epsilon=0.001$. For the SIR particle filter parameters: $\eta=50$, $\chi=50$,

$\theta=3$, number of segments=3. The control law sample time Δt is set to 40ms.

B. Performances evaluation criteria

The method is evaluated with regards to several criteria: the average displacement d_{av} , the average error e_{Av} , the relative norm error \overline{e}_n , the median duration of computation process t_{med} .

The average displacement between the undeformed wire shape and the one at the final discrete time k_f is defined by:

$$d_{av} = \frac{\sum_{i=1}^{\eta} \|\tilde{\mathbf{p}}_i(k_f) - \tilde{\mathbf{p}}_i(0)\|}{\eta} \quad (15)$$

The average error is the average of the Euclidean distance between a current wire shape point and its desired position. It is defined by:

$$e_{Av}(k) = \frac{\sum_{i=1}^{\eta} \|\tilde{\mathbf{p}}_i(k) - \tilde{\mathbf{p}}_i^*\|}{\eta} \quad (16)$$

The relative norm error, expressed in percentage, is defined by:

$$\overline{e}_n = \frac{100}{\eta} \sum_{i=1}^{\eta} \frac{\|\tilde{\mathbf{p}}_i^* - \hat{\tilde{\mathbf{p}}}_i(0)\| - \|\hat{\tilde{\mathbf{p}}}_i(k_f) - \hat{\tilde{\mathbf{p}}}_i(0)\|}{\|\tilde{\mathbf{p}}_i^* - \hat{\tilde{\mathbf{p}}}_i(0)\|} \quad (17)$$

This error expresses the ratio of the difference between the expected deformation and the final one.

The median duration is the duration required to perform the tracking of the wire, the update of the Jacobian and the computation of the control law. It is defined by:

$$t_{med} = \text{median}_{k=0}^{k_f} (\Delta t(k)) \quad (18)$$

The convergence duration presented in Tables I and II corresponds to the duration that was required to make the shape of the wire converge towards the desired one.

C. Influence of the type of robots actions

We first evaluate the performances of the method with regards to the robots actions responsible for the deformation of the wire. Shape servoing involving translation only, rotation only and a combination of rotations and translations of both end-effectors have been tested. Table I presents the results of these experiments. The method was able to converge below 5mm in terms of average error e_{Av} and below 13% in terms of the relative norm error \overline{e}_n for every translation-based experiment. It was harder to notice deformations using only X-axis and Z-axis rotation-based motions of the end-effectors because the resulting deformation was mainly occurring in the Z axis of the camera frame. The measurements in this direction are approximate due to the quality of the RGB-D sensor. Consequently, the relative norm error is greater for the rotation-based experiments. For the combined rotations-translations experiments, the method was able to converge to 2.8mm in terms of average error and below 12% in terms of the relative norm error. In almost all experiments, the average displacement is lower than the average initial error. It indicates that the method did not over-deform the wire during the experiments. For all the experiments, the median duration of computation process was around 20ms, proving that the method can work at an interactive time.

TABLE I: Results of experiments where the deformation resulted from different robots DOF motions. The four first columns give the average displacement, the relative norm error, the initial average error and the final average error respectively. The fifth column indicates the duration required to make the wire converge towards the desired shape. The last column indicates the robots actions that generated the desired deformed shape. “tx”, “ty” and “tz” correspond to translations along each axis, while “rx”, “ry” and “rz” correspond to rotations. The last three lines correspond to combination of translations and rotations.

d_{av} (mm)	\bar{e}_n (%)	$e_{Av}(0)$ (mm)	$e_{Av}(k_f)$ (mm)	Convergence duration (s)	Robots actions
17.6	9.3	17.5	2.5	70	tx
10.5	10.6	10.3	2.7	170	ty
11.7	12.2	12.2	3.6	70	tz
12.9	27.3	11.2	4.7	190	rx
11.6	15.7	11.6	2.1	70	ry
9	20.6	8	4.1	50	rz
13.2	12.9	15.1	4.7	150	txtytz
16.2	8.6	15.5	2.8	130	rztxtytz
12.6	11.4	12.7	2.8	70	rxryrztxtytz

TABLE II: Evaluation criteria with regards to the resolution of the B-Spline. The four first columns are the same as in Table I. The fifth column indicates the median duration of computation process. The sixth column indicates the convergence duration defined as in Table I. The seventh column indicates the number of points used to model the wire. For each experiment, a combination of translations along every axis of the robots has been performed, as shown in the last column.

d_{av} (mm)	\bar{e}_n (%)	$e_{Av}(0)$ (mm)	$e_{Av}(k_f)$ (mm)	t_{med} (ms)	Convergence duration (s)	η	Robots actions
13.4	16.0	14.6	3.5	3.1	100	10	txtytz
15.3	13.2	15.2	3.2	5.6	140	25	txtytz
13.5	5.5	13.7	4.6	10.3	100	50	txtytz
13.3	16.2	15.8	5.5	14.8	90	75	txtytz
15.5	18.9	16	3.1	19.8	100	100	txtytz

D. Influence of the wire material

We tested the independence of the method from the mechanical properties and visual properties of the wire. We performed several experiments using wires of different materials, diameters and colors: two twined elastic ropes of eight-millimeter in different colors and a 13-millimeter flexible pale grey gas pipe. The different objects are presented in Fig. 2 as insets. The experiment combined deformations due to the rotation along the Z axis and translations along every axis of both end-effectors coordinates frame. Figures 3a to 3c illustrate the final shape of the deformable wire projected on the XY, XZ, YZ planes in order to observe the 3D deformation for each of these experiments, as well as the evolution of the average error over time. In these figures, the relative norm error is coded according to a color scale: the greener the wire the closer to 0% the error is, the redder the wire the closer to 30% the \bar{e}_n error is. From similar initial and desired shapes, the method was able to converge with a similar accuracy: the final average error is 3.3mm for the red wire, 4mm for the green wire and 2.9mm for the gas pipe. However, one can notice in the different plan projections that the gas pipe converged to a smoother shape than the elastic ropes. The relative norm error is also lower in the gas pipe experiment: 3.9% against 10.6% for the red wire and 9.4% for the green wire. The material of which it is composed may have induced smoother deformations that are therefore easier to control than the twined ropes. The median duration of computation process was around 17ms for each wire.

E. Influence of the resolution of the B-Spline

We also performed tests to determine if the accuracy of the method is related to the number of points η of the B-spline used to model the shape of the wire. Experiments have been conducted combining translations along every axis of both end-effectors coordinates frame. The same initial and desired shapes were used for each of these experiments, which were performed using $\eta = 10, 25, 50, 75$ and 100 points respectively. Table II presents the results of these experiments. Starting with similar initial and desired shapes, the method was able to converge with a similar accuracy: the final average error is 3.5mm for 10 points, 3.2mm for 25 points, 4.6mm for 50 points, 5.5mm for 75 points and 3.1mm for 100 points. In terms of time performance, the method was still working at an interactive time using $\eta = 100$, with a median duration of computation process of 19.8ms.

F. Large deformation experiments

The deformation Jacobian is only valid for a small range of wire shape deformations. We carried out an experiment to determine if it could prevent the method to be used to generate large deformations. The initial shape of the wire is depicted as inset in Figure 2, while the desired shape is depicted in Figure 4a. Figure 5 shows the evolution of the average error between the current shape and the desired one, as well as the initial, final and desired shapes of the wire displayed in the XY, XZ and YZ planes. The final shape of the wire uses the same color-coded representation as in IV-D. One can notice that the method failed to converge towards the desired final

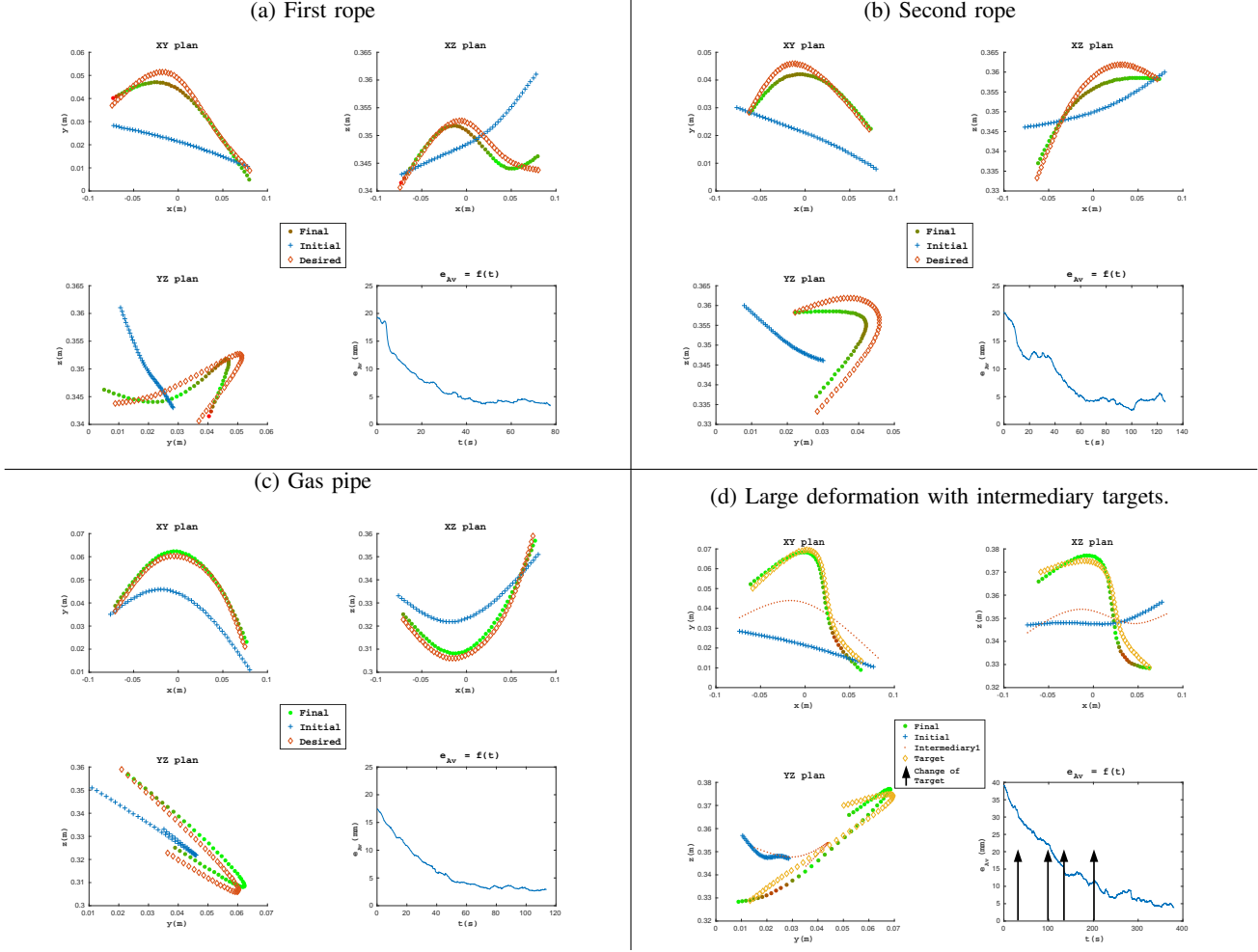


Fig. 3: Initial, final and desired shapes displayed in different planes and evolution of the average error over time for the manipulations of three different wires (TL, TR: two ropes with different colors; BL: higher diameter wire). Figure 3d presents one of the intermediary targets used in Section IV-G along with the moments at which the target was changed.

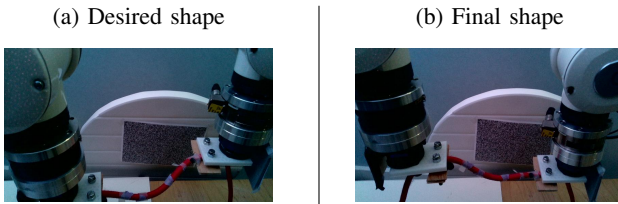


Fig. 4: Experiment with large deformations of the wire.

shape. This is due to the fact that it fell into a local minima at time $t=25s$. This motivated us to conduct experiments aiming at generating large deformations by considering a succession of desired intermediary shapes.

G. Adaptation of the method to handle large deformations

To enable our method to reach a desired deformation that differs too largely from the initial state, we can define successive intermediary targets. We carried out an experiment using this approach by defining before the experiment a trajectory composed of four desired intermediary shapes and a desired final shape. The intermediary shapes are chosen between the initial and desired shapes such as they describe a sequence

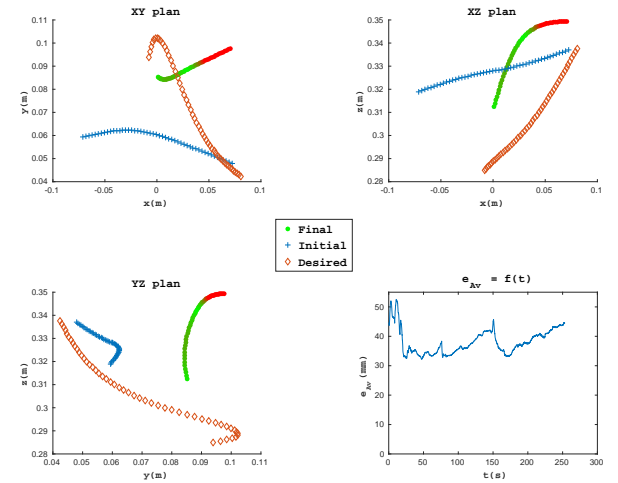


Fig. 5: Large deformation case: initial, final and desired shapes shown in different planes and average error over time.

from the furthest to the closest shape of the final desired shape, as depicted in Figure 6. The overall experiment lasted 380s. The median duration of computation process was 16.6ms. The average displacement between the initial and final shape was

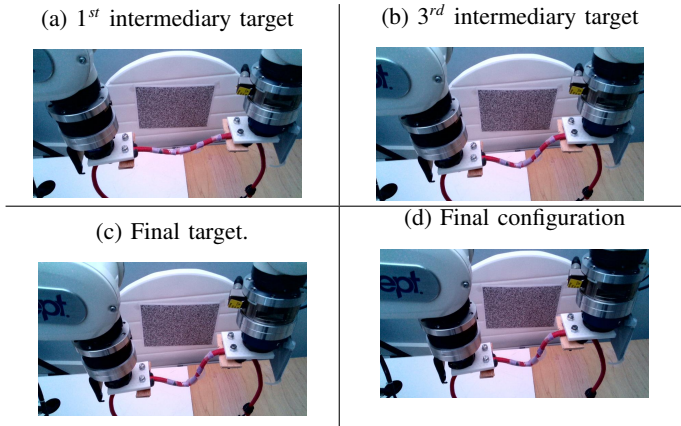


Fig. 6: Intermediary targets for reaching large deformations of the wire. Three first pictures: some of the intermediary targets. Last picture: final target and final configuration reached at convergence.

39.3mm. The initial average error was $e_{Av}(0) = 39\text{mm}$ while the final one was $e_{Av}(k_f) = 3.9\text{mm}$. Fig. 3d shows the evolution of the average error between the current shape and the desired final one, as well as the initial, final and desired shapes of the wire displayed in the XY, XZ and YZ planes. The final shape of the wire uses the same color-coded representation as in IV-D. It is possible to notice that the average error between the current shape and the final desired one decreased regularly during the experiment. The change of intermediary targets did not disturb this convergence. It is possible to notice that the relative norm error is greater on an extremity of the wire where the torsion was the most important. However, the relative norm error is only 6.1%, which proves that the method has converged well towards the desired final shape.

V. CONCLUSION

In this paper, we addressed the challenge of the automatic 3D shape control of deformable wires. First, we proposed a novel visual feature that permits to model the 3D shape of a deformable wire. This visual feature relies on a geometric model and the use of SIR particle filtering. Second, automatic shape control is ensured by an adaptive model-free shape servoing method. Controlling two end-effectors that hold both extremities of the wire, the method permits to make the wire reach a desired 3D shape. Experiments demonstrated that the method could be used to reach complex 3D shapes by defining intermediary targets. As future work, it would be interesting to automatically generate these intermediary targets from the initial and desired shapes.

REFERENCES

- [1] P. Chatelain, A. Krupa, and N. Navab, "3d ultrasound-guided robotic steering of a flexible needle via visual servoing," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 2250–2255.
- [2] T. Behr, T. P. Pusch, M. Siegfarth, D. Hüsener, T. Mörschel, and L. Karstensen, "Deep reinforcement learning for the navigation of neurovascular catheters," *Current Directions in Biomedical Engineering*, vol. 5, no. 1, pp. 5–8, 2019.
- [3] S. B. Kesner and R. D. Howe, "Force control of flexible catheter robots for beating heart surgery," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 1589–1594.
- [4] G. Fang, X. Wang, K. Wang, K.-H. Lee, J. D. Ho, H.-C. Fu, D. K. C. Fu, and K.-W. Kwok, "Vision-based online learning kinematic control for soft robots using local gaussian process regression," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1194–1201, 2019.
- [5] J. Zhu, B. Navarro, R. Passama, P. Fraithe, A. Crosnier, and A. Cherubini, "Robotic manipulation planning for shaping deformable linear objects with environmental contacts," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 16–23, 2019.
- [6] M. Laranjeira, C. Dune, and V. Hugel, "Catenary-based visual servoing for tethered robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 732–738.
- [7] A. Schlechter and D. Henrich, "Manipulating deformable linear objects: Manipulation skill for active damping of oscillations," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 2, 2002, pp. 1541–1546.
- [8] F. Ding, J. Huang, Y. Wang, T. Fukuda, and T. Matsuno, "Adaptive sliding mode control for manipulating deformable linear object with input saturation," in *Proc. of the IEEE Int. Conf. on Mechatronics and Automation*, 2012, pp. 1862–1867.
- [9] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.
- [10] Z. Hu, P. Sun, and J. Pan, "Three-Dimensional Deformable Object Manipulation Using Fast Online Gaussian Process Regression," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 979–986, 2018.
- [11] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3426–3433, 2018.
- [12] M. Verghese, F. Richter, A. Gunn, P. Weissbrod, and M. Yip, "Model-free visual control for continuum robot manipulators via orientation adaptation," *arXiv preprint arXiv:1909.00450*, 2019.
- [13] P. Qi, C. Liu, L. Zhang, S. Wang, H.-K. Lam, and K. Althoefer, "Fuzzy logic control of a continuum manipulator for surgical applications," in *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics*, 2014, pp. 413–418.
- [14] J. Zhu, B. Navarro, P. Fraithe, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 479–484.
- [15] T. Matsuno and T. Fukuda, "Manipulation of flexible rope using topological model based on sensor information," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 2638–2643.
- [16] F. Abegg and H. Worn, "Robust algorithms for recognizing shape changes of deformable linear objects in video image sequences," in *Proc. of the IEEE Int. Conf. on Image Processing*, vol. 1, 2000, pp. 335–338.
- [17] N. Padoy and G. D. Hager, "Deformable tracking of textured curvilinear objects," in *Proc. of British Machine Vision Conference*, 2012, pp. 1–11.
- [18] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration," *The International Journal of Robotics Research*, pp. 1–16, 2018.
- [19] C. Chi and D. Berenson, "Occlusion-robust deformable object tracking without physics simulation," in *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, 11 2019, pp. 6443–6450.
- [20] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *Proc of the IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 1130–1137.
- [21] C. De Boor, "On calculating with b-splines," *Journal of Approximation theory*, vol. 6, no. 1, pp. 50–62, 1972.
- [22] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *Proc. of the IEE Conf. F (radar and signal processing)*, vol. 140, no. 2, 1993, pp. 107–113.
- [23] R. Lagneau, A. Krupa, and M. Marchal, "Active deformation through visual servoing of soft objects," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2020.
- [24] Intel. Tuning depth cameras for best performance. [Online]. Available: dev.intelrealsense.com/docs/tuning-depth-cameras-for-best-performance
- [25] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011.
- [26] É. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.